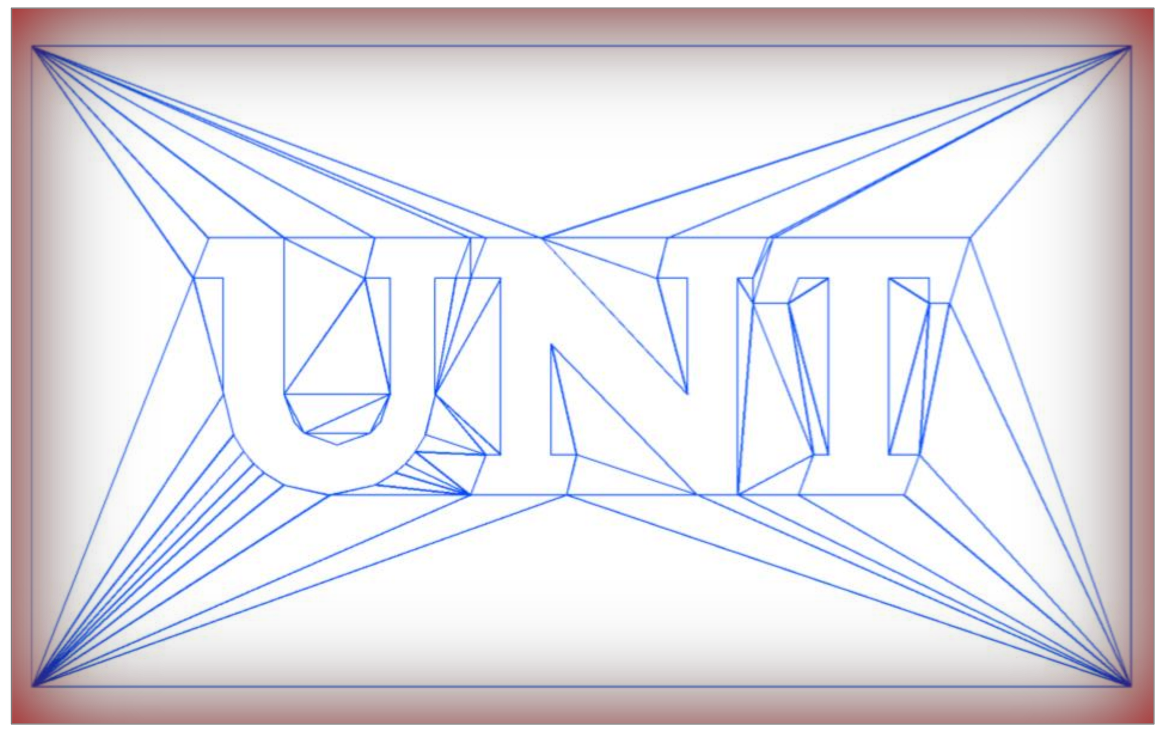
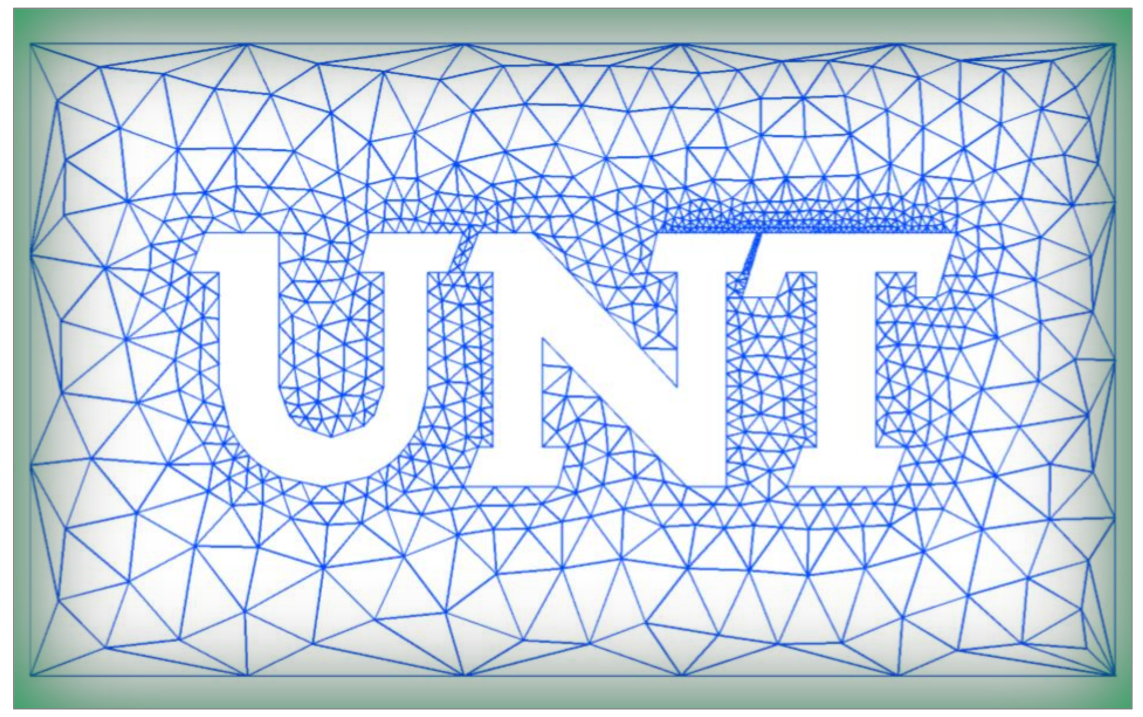
Triangle mesh generation combining edge splitting and angle-based smoothing

Samuel B Johnson, samuelbjohnson@live.com -- Robert J Renka, robert.renka@unt.edu -- University of North Texas Department of Computer Science -- 2016

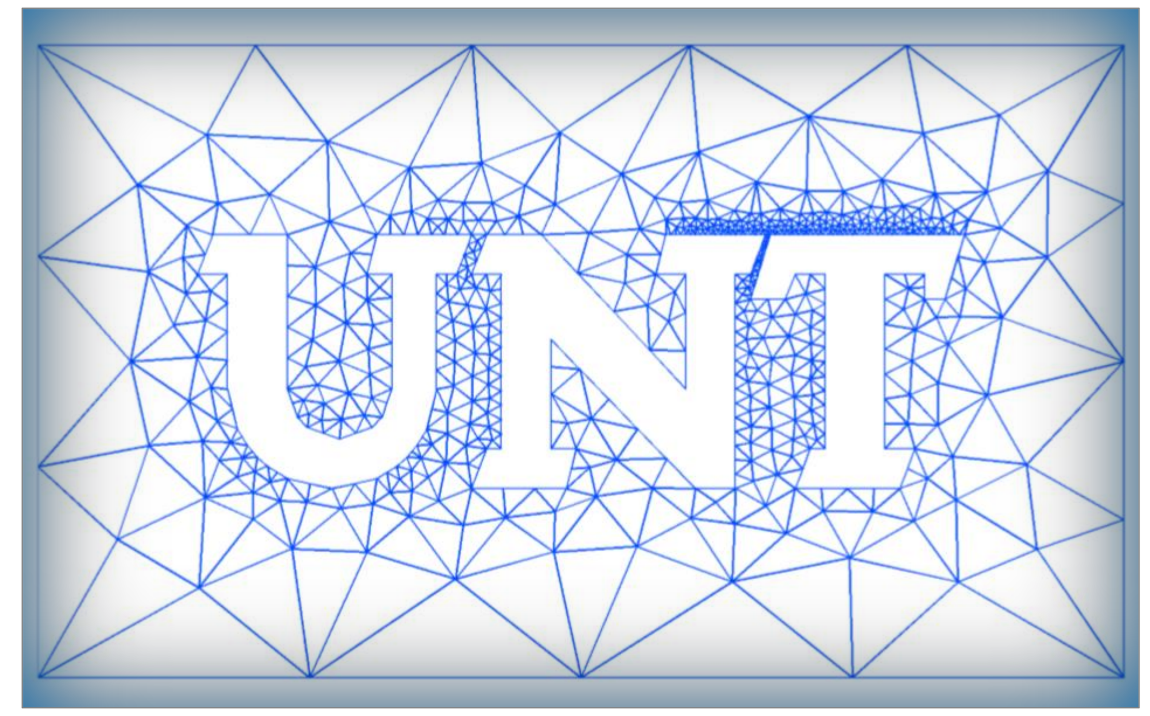
Original Mesh



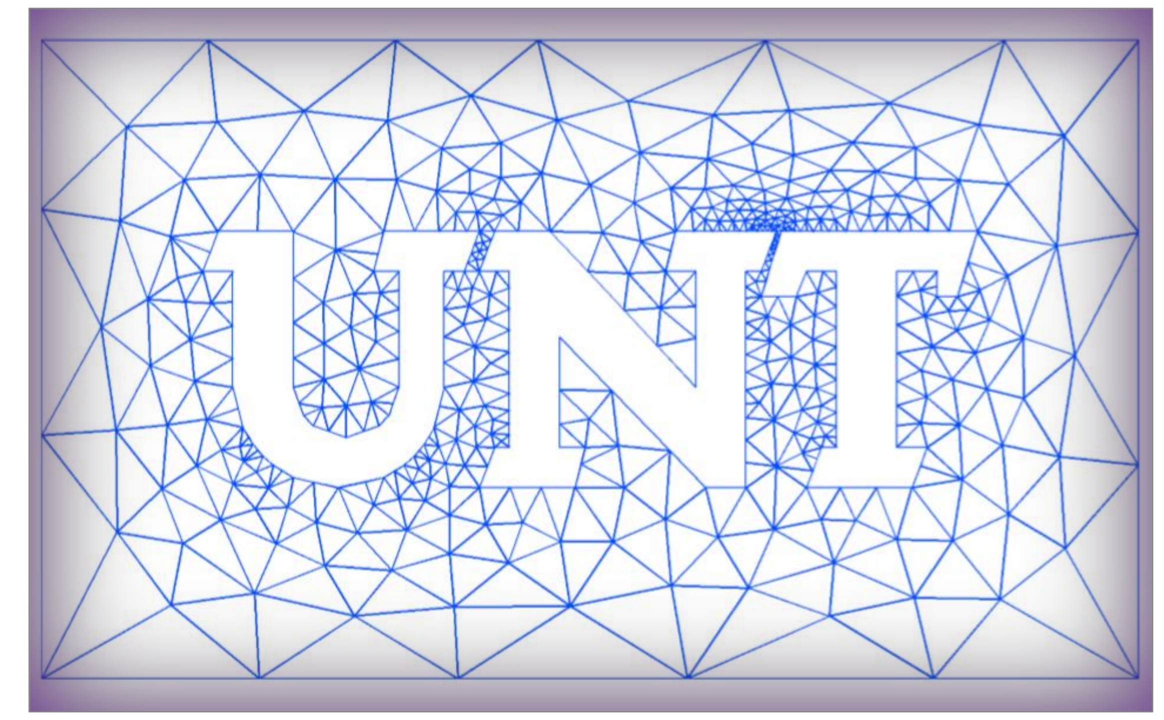
CVT (Slow) Method

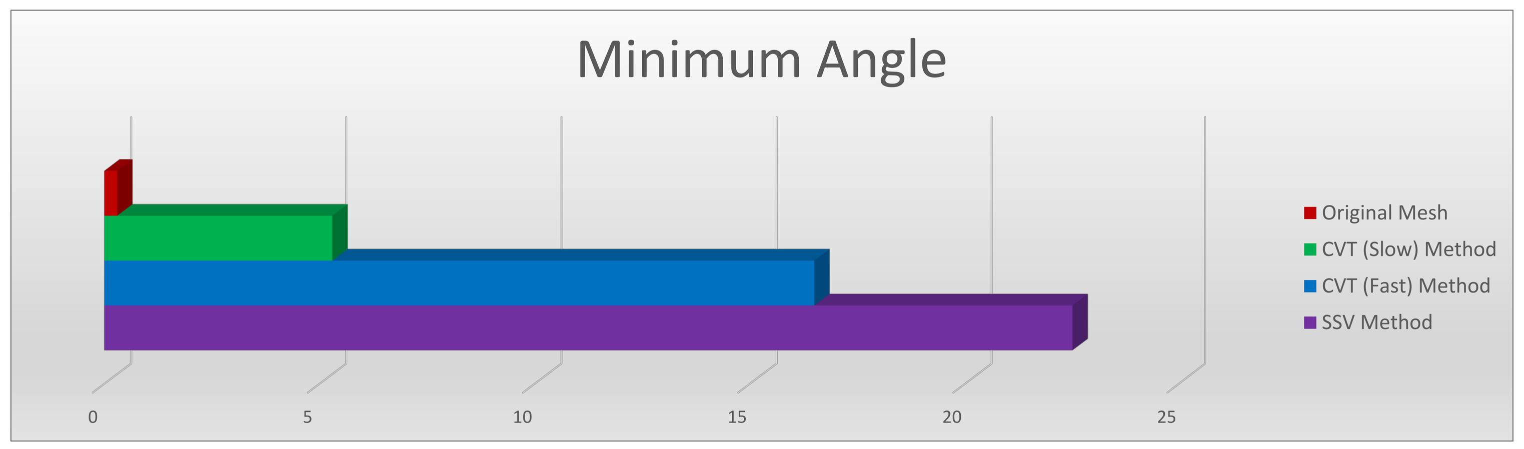


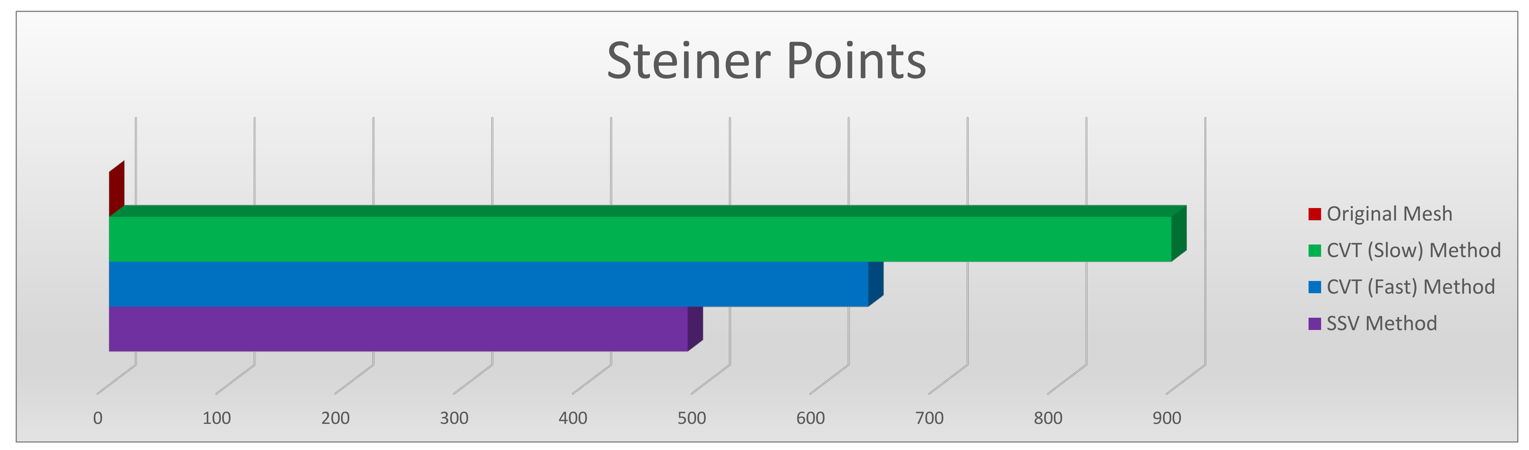
CVT (Fast) Method

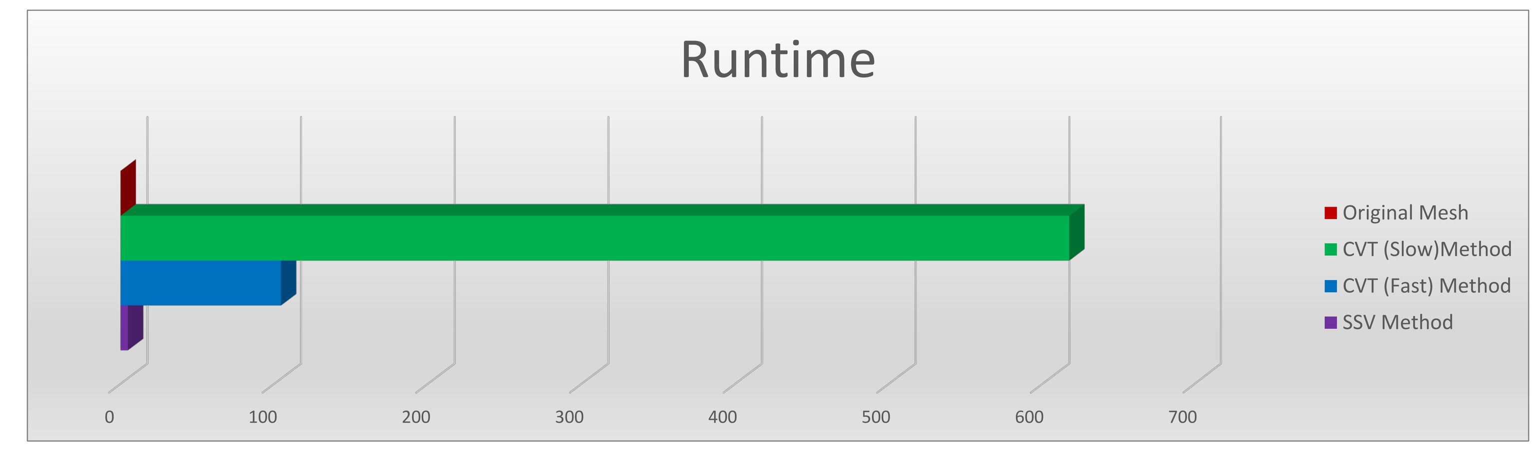


New SSV Method









ALGORITHM

- We consider the problem of generating meshes with high triangle quality and as few added vertices as possible, while satisfying constraints on size and geometry.
- We propose a simple mesh generation method which alternates between:
  - A **refinement step** involving long edge bisection
  - An **optimization step** involving Delaunay edge swaps and a new, efficient angle-based smoothing procedure

SSV mesh generation algorithm

Input: A coarse planar triangular mesh T constrained within the bounded domain $\Omega \subset \mathbb{R}^2$.
Let Ω be defined by the fixed boundary vertices and edges of T, and T may include zero or more free interior vertices.
Let μ be a provided sizing function.
Repeat
Flag boundary and isolated interior edges whose size ratios (the ratio of the length of an edge to the μ value at its midpoint) exceed the target size ratio (half the maximum size ratio found in the mesh in a given refinement step).
Isolated edges are those whose size ratio exceeds that of all neighboring edges.
Sort flagged edges in order of descending size ratio.
Repeat
Refinement of T by inserting a Steiner point at the midpoint of the next flagged edge.
Repeat
Optimization of T by first applying local interior edge swaps as necessary to enforce the Delaunay angle criterion, then applying local angle-based SSV smoothing sweeps.
Until Stopping criterion: No edge swaps were required and the maximum displacement of a vertex (relative to its μ value) was lower than a given tolerance t_{gs}.
Until Stopping criterion: The refinement step has processed all flagged edges.
Until Stopping criterion: All edges satisfy the sizing constraint determined by μ, or the number of Steiner points added exceeds an optional provided Steiner point constraint.
Output: Final triangle mesh.

COMPARISON

- We compare our method, which we label the SSV Method, to the original input mesh, as well as the following 2 methods:
  - The method originally described by Tournois et al. (2007)[1], which we label as the CVT (Slow) Method
  - A faster variant of this method we developed which avoids the ill-conditioned problem of computing a circumcenter of a nearly null triangle, which we label as the CVT (Fast) Method
- Results from a wide variety of test cases demonstrate that, compared to the original CVT (Slow) Method and the CVT (Fast) variant, our method consistently results in
  - **Comparable quality measures**
  - **Fewer Steiner points**
  - **Significantly faster runtime**
- One of these test cases, using an input coarse mesh on a bounded domain defined by a PSLG of the UNT logo and a variable mesh width sizing constraint with k = 0.75 (method described in Tournois et al. (2007)[1]), is shown above. Compared to CVT (Slow) in this test case, the SSV method results in:
  - Improvement in **minimum angle**, a metric of triangle quality, from 5.3 to 22.5 degrees
  - Reduction in **number of Steiner points** from 894 to 487 points
  - Significantly reduced **runtime** by 99.2%
- The tests were run on a MacBook Air with a 1.8 GHz i7 processor running Matlab R2013b.

RESULTS

- In the table below, we show results from some more of our test cases. These statistics are representative of the performance of our method on all 36 cases.

Mesh	Width	$n_{v(start)}$	Method	α	n_{sp}	t
M shape	Constant 0.05	13	None	7.9	0	0
			CVT (Slow)	36.3	828	54.6
			CVT (Fast)	25.5	750	74.3
			SSV	31.6	745	4.8
UNT logo	Constant 0.20	70	None	0.3	0	0
			CVT (Slow)	16.6	3669	345.3
			CVT (Fast)	13.1	3688	252.3
			SSV	23.4	3226	22.9
77-vertex complex boundary mesh	Constant 0.05	77	None	0.6	0	0
			CVT (Slow)	18.7	2669	476.6
			CVT (Fast)	22.0	2691	295.6
			SSV	21.5	2659	20.1
77-vertex complex boundary mesh	Variable k = 0.75	77	None	0.6	0	0
			CVT (Slow)	17.0	615	99.4
			CVT (Fast)	16.7	520	60.1
			SSV	17.6	479	4.9

$n_{v(start)}$ = Number of vertices in starting mesh
α = Minimum Angle
n_{sp} = Number of Steiner points
t = Runtime

REFERENCES
[1] Tournois J, Alliez P, Devillers O. Interleaving Delaunay refinement and optimization for 2D triangle mesh generation. Proceedings, 16th International Meshing Roundtable, Springer-Verlag, October 14-17 2007, 83–101.